

**NPS ARCHIVE**  
**1969**  
**ALLEN, R.**

A DIGITAL COMPUTER PROGRAM FOR DETERMINING  
THE FREQUENCY RESPONSE OF A LINEAR SYSTEM  
TO A SINUSOIDAL EXCITATION

by

Richard Owen Allen, Jr.

**DUDLEY KNOX LIBRARY  
NAVAL POSTGRADUATE SCHOOL  
MONTEREY, CA 93943-5101**

# United States Naval Postgraduate School



## THE SIS

A DIGITAL COMPUTER PROGRAM FOR DETERMINING  
THE FREQUENCY RESPONSE OF A LINEAR SYSTEM  
TO A SINUSOIDAL EXCITATION

by

Richard Owen Allen, Jr.

December 1969

*This document has been approved for public re-  
lease and sale; its distribution is unlimited.*

LIBRARY  
NAVAL POSTGRADUATE SCHOOL  
MONTEREY, CALIF 93940

A Digital Computer Program for Determining the Frequency  
Response of a Linear System to a Sinusoidal Excitation

by

Richard Owen Allen, Jr.  
Commander, United States Navy  
B.S., United States Naval Academy, 1956

Submitted in partial fulfillment of the  
requirements for the degree of

MASTER OF SCIENCE IN ELECTRICAL ENGINEERING

from the

NAVAL POSTGRADUATE SCHOOL  
December 1969

~~File~~ A 3793-2.1

NPS Archive

1969

Apr, R

ABSTRACT

The application of digital computers for the simulation of physical systems has become widespread. This paper describes a program designed to measure the frequency response of the simulation model of a linear system.

TABLE OF CONTENTS

I. INTRODUCTION . . . . . 7

II. DISCUSSION . . . . . 9

    A. SYSTEM RESPONSE TO A SINUSOIDAL INPUT . . . . . 9

    B. RESONANCE IN A SYSTEM AND STABILITY . . . . . 10

    C. DEVELOPMENT OF THE PROGRAM . . . . . 10

    D. ASSUMPTIONS . . . . . 10

    E. DETERMINATION OF THE STEADY-STATE CONDITION . . . . . 11

    F. DETERMINATION OF OUTPUT PEAK MAGNITUDE AND PHASE . . . . . 12

III. THE PROGRAM . . . . . 13

    A. THE SYSTEM . . . . . 13

    B. CONSTANTS, PARAMETERS, AND PROGRAM CONTROLS . . . . . 13

    C. THE INITIAL REGION . . . . . 16

    D. THE DERIVATIVE REGION . . . . . 16

    E. THE DYNAMIC REGION . . . . . 17

    F. THE TERMINAL REGION . . . . . 20

IV. CONCLUSIONS . . . . . 21

    A. RESULTS . . . . . 21

    B. LIMITATIONS . . . . . 21

    C. RECOMMENDATIONS FOR EXTENSIONS OF INVESTIGATION . . . . . 22

APPENDIX A - EXAMPLES . . . . . 24

APPENDIX B - PROGRAM FOR GENERATING TABLE OF DELT VERSUS OMEGA . . . . . 44

LIST OF REFERENCES . . . . . 45

INITIAL DISTRIBUTION LIST . . . . . 46

FORM DD 1473 . . . . . 47





## LIST OF ILLUSTRATIONS

<u>Figure</u>		<u>Page</u>
3.1	Program for Determining Steady-state Response to a Sinusoidal Excitation	14
A.1	Program for System of Example 1	26
A.2	Frequency Response for System of Examples 1 - 3	28
A.3	Program for System of Example 2	30
A.4	Program for System of Example 3	32
A.5	Program for System of Example 4	34
A.6	Frequency Response for System of Example 4	35
A.7	Block Diagram of System for Example 5	36
A.8	Program for System of Example 5	37
A.9	Frequency Response for System of Example 5	38
A.10	Diagram of System for Example 6	40
A.11	Program for System of Example 6	41
A.12	Frequency Response for System of Example 6	43
B.1	Program for Generating Table of DELT versus OMEGA	44

## ACKNOWLEDGEMENT

The author wishes to thank Professor George J. Thaler for originally suggesting the problem and for his continuous encouragement and guidance throughout this project.

## I. INTRODUCTION

The application of digital computers for the simulation of physical systems has become widespread. Computer programs have been developed that simulate mechanical, electrical, and electro-mechanical systems, circuits, chemical processes, biomedical problems, traffic control, and so on.

One of the most important groups of programs for the simulation of continuous systems is that which employs digital-analog simulators; i.e., programs which simulate the elements and organization of the analog computer.

The purpose of the investigation presented in this paper is to develop a technique -- a digital computer program -- for measuring the frequency response of the simulation model of a system. While such a program should be applicable to any digital simulation language, the program description presented in this paper utilizes International Business Machines Company's DSL/360 Digital Simulation Language. DSL/360, a System/360 FORTRAN IV program for the digital simulation of continuous system dynamics, employs the building-block approach of digital-analog simulators while providing the power of logical and algebraic equation notation.

DSL/360 provides a basic set of function blocks from which a physical system may be modeled: integrators, limiters, pulse generators, function generators, and so on. In addition, FORTRAN library functions and functions from the Scientific Subroutine Package (SSP) may be utilized. In the event none of these satisfies the user's needs, the user may provide his own function blocks or subroutines.

A digital computer program to determine the steady-state response of a system to a sinusoidal input is a useful adjunct to a digital simulation language. With it, non-linearities in an ostensibly linear system may be detected and evaluated.

Reference 1 describes in detail the use and operation of DSL/360.

## II. DISCUSSION

This chapter contains a discussion of the response of a linear system to a sinusoidal excitation and the basic concepts utilized in the development of a digital computer program to determine such response.

### A. SYSTEM RESPONSE TO A SINUSOIDAL INPUT

When a system is linear, its response to a sinusoidal input is a sine wave of the same frequency (the higher harmonics are negligible). The magnitude ratio and phase of the response depend on the forcing frequency but not on the input magnitude [Ref. 2]. The condition that magnitude ratio and phase be independent of the input amplitude is a condition for the linearity of the system.

The steady-state frequency response of a stable, linear system to a sinusoidal input can be determined analytically from the system transfer function [Ref. 3]. The response to an input  $A \sin \omega t$  is given by

$$y = A |P(j\omega)| \sin(\omega t + \Phi)$$

where  $|P(j\omega)|$  is the magnitude of  $P(j\omega)$ ,  $\Phi$  is the argument of  $P(j\omega)$ , and the complex number  $P(j\omega)$  is determined from the system transfer function  $P(s)$  by replacing the  $s$  with  $j\omega$ . The system output has the same frequency as the input and can be obtained by multiplying the input by  $P(j\omega)$  and shifting the phase angle of the input by the argument of  $P(j\omega)$ .  $|P(j\omega)|$  and  $\Phi$  for all  $\omega$  constitute the system frequency response, where  $P(j\omega)$  is the gain of the system for sinusoidal inputs with frequency  $\omega$ .

## B. RESONANCE IN A SYSTEM AND STABILITY

The maximum value of the magnitude of the closed-loop frequency response of a system is a measure of the stability of the system [Ref. 3]. The frequency at which this maximum occurs is the resonant frequency of the system.

It often occurs in reality that a linear system's response will contain non-linearities under certain conditions. As an example, a second-order system may experience system gain non-linearity as a result of resonance if the excitation frequency is at or near the system's natural frequency. Such non-linearity and any resultant effect on system stability are of interest to a design engineer, for instance, since the performance of the system with which he is concerned may be vitally affected. Being able to utilize a computer to determine system non-linearity during the design stage may result in savings of both time and money.

## C. DEVELOPMENT OF THE PROGRAM

A program to determine the frequency response of a system to a sinusoidal input must first determine attainment of the steady-state condition upon application of the excitation and then calculate values of the steady-state response magnitude and phase.

## D. ASSUMPTIONS

Several assumptions are made regarding the system, its excitation, and its response.

Assumption 1. The system is linear and stable.

Assumption 2. The system may be described in DSL/360 format.

Assumption 3. There are no initial conditions; i.e., all initial conditions are zero at time zero.

Assumption 4. The excitation is described by

$$A \sin(\omega t + \theta)$$

where A is the peak amplitude,  $\omega$  is the angular frequency in radians per second, and  $\theta$  is the phase (input phase is zero throughout this paper).

Assumption 5. The steady-state output of the system is periodic with the same frequency as the input and may be described by

$$B \sin(\omega t + \phi)$$

where B is the peak amplitude,  $\omega$  is the angular frequency in radians per second, and  $\phi$  is the phase of the output with respect to the input.

#### E. DETERMINATION OF THE STEADY-STATE CONDITION

In any digital simulation language, the independent variable is time. The computational process is an iterative one, performed at intervals of time specified by the user. Attainment of the steady-state condition is determined by a mechanical, value-comparing process rather than by the solution of an equation, such as that discussed earlier.

The steady-state condition is determined in the program discussed in this paper by permitting the system to respond to the sinusoidal input until certain steady-state criteria are met. Ten cycles of input are permitted to pass before attempting any steady-state testing, by-passing computations involving the more wildly fluctuating transients and, thereby, decreasing computation time. The figure "ten" is an arbitrary choice, based on the assumption that the output will have settled to a condition approaching the steady state after ten cycles of input.

The steady-state determination is accomplished by comparing the value of the output magnitude at each iteration point on one cycle with the value at the same point on a later cycle. When these values are reasonably close, steady state is considered to have been attained. To reduce the possibility of chance satisfaction of the steady-state criteria while the output is still in the transient or settling phase, comparison of output magnitude values is made in the program with not one but two later cycles, the third and the tenth. Again, these figures are arbitrary choices.

#### F. DETERMINATION OF OUTPUT PEAK MAGNITUDE AND PHASE

Once the steady-state testing commences, the program simultaneously computes the peak magnitude of the output and its phase with respect to the input.

The peak magnitude of the output is nothing more than the highest steady-state value computed for the output magnitude.

The phase is computed from the times at which an input cycle and its resultant output cycle cross their respective zero-value points.

The equations used in the program are:

$$\phi_R = \frac{2\pi(t_i - t_o)}{T}$$

$$\phi_D = \frac{360(t_i - t_o)}{T}$$

where  $\phi_R$  is the phase in radians,  $\phi_D$  is the phase in degrees,  $t_i$  is the time at which the input cycle crosses its zero-value point,  $t_o$  is the time at which the resultant output cycle crosses its zero-value point, and  $T$  is the period of both the input and the output.



### III. THE PROGRAM

This chapter presents a description of the program evolved to determine the response of a linear system to a sinusoidal excitation. A sample program of a single-run job is shown in Figure 3.1. Reference to this program is made in the explanations in the paragraphs that follow. The explanations presuppose some familiarity with DSL/360 on the part of the reader.

#### A. THE SYSTEM

The system used in the sample program of Figure 3.1 is second-order, with a natural frequency of ten radians per second, but any linear system would have served as well. The system is described, from its closed-loop form, as having a forward-loop transfer function of  $100/s(s + 1)$  and a negative unity feedback-path transfer function.

The system might also have been modeled in its open-loop form. The system modeled in this manner is shown in Appendix A.

#### B. CONSTANTS, PARAMETERS, AND PROGRAM CONTROLS

Constants, parameters, and program execution controls are shown in the sample program following the title and general system description (in DSL/360, asterisks in column 1 indicate a comment card).

The one constant, PI, is self-explanatory.

The parameters must be determined by the user and are:

OMEGA - the input frequency, in radians per second

A----- the input peak magnitude

FGAIN - the feedback-path gain

```

TITLE    LINEAR SYSTEM RESPONSE TO A SINE WAVE INPUT
*        FORWARD-PATH TRANSFER FUNCTION: 100/S(S+1)
*        FEEDBACK-PATH TRANSFER FUNCTION: -1
CONST   PI=3.1415927
PARAM   OMEGA=20.,A=1.,FGAIN=1.
CONTRL  FINTIM=1000.,DELT=0.015707964
INTEG   RKSF
INITIAL REGION
        PERIOD=2.*PI/OMEGA
        DELINT=3.*PERIOD
        DILINT=10.*PERIOD
        ERG=PI*(.5-DELT/PERIOD)
        EPS1=1.E-03*A
        EPS2=0.
        EPS3=0.
        XIN=0.
        XOUT=0.
        XTDIFF=0.
        RPHASE=0.
        DPHASE=0.
        MAXOUT=0.
        DEROUT=10.
DERIVATIVE REGION
        IN=A*SINE(C.,OMEGA,0.)
        ERPOP=IN-FGAIN*OUT
        OUT=TRNFR(0.,2.,IC,NUM,DEN,ERROR)
STORAG  IC(2),NUM(1),DEN(3)
TABLE  IC(1-2)=0.,0.,NUM(1)=100.,DEN(1-3)=1.,1.,0.
DYNAMIC REGION
* DETERMINATION OF STEADY-STATE CONDITION
        DELOUT=DELAY(605,DELINT,OUT)
        DILOUT=DELAY(2005,DILINT,OUT)
        IF(TIME.LE.DELINT) GO TO 2
        IF(ABS(DELOUT-OUT).LE.EPS1) GO TO 1
        MAXOUT=C.
        GO TO 2
* DETERMINATION OF OUTPUT STEADY-STATE PEAK MAGNITUDE
        1 IF(OUT.GT.MAXOUT) MAXOUT=OUT
        DEROUT=OMEGA*MAXOUT*COS(ERG)
        EPS2=0.001*MAXOUT
        EPS3=OMEGA*MAXOUT*COS(ERG)
* DETERMINATION OF OUTPUT STEADY-STATE PHASE
        2 AXIN=CROSS(TIME,IN,0.)
        IF(AXIN.NE.0.) XIN=AXIN
        AXOUT=CROSS(TIME,OUT,0.)
        IF(AXOUT.NE.0.) XOUT=AXOUT
        IF(XIN.GT.XOUT) GO TO 3
        XTDIFF=XIN-XOUT
        RPHASE=2*PI*XTDIFF/PERIOD
        DPHASE=360.*XTDIFF/PERIOD
        3 ARG=OMEGA*TIME+RPHASE
        IF(TIME.LE.DILINT) GO TO 4
        IF(ABS(DELOUT-OUT).LE.EPS2.AND.ABS(DILOUT-OUT...
        ).LE.EPS2.AND.ABS(DEROUT...
        ).LE.EPS3.AND.OUT.GT.0..AND.MAXOUT.GT.0.) CALL ENDRUN
        4 CONTINUE
TERMINAL REGION
PRINT  OMEGA,MAXOUT,RPHASE,DPHASE
CALL  PRINT
END
STOP

```

FIGURE 3.1

SAMPLE PROGRAM FOR DETERMINING SYEADY-STATE RESPONSE TO A SINUSOIDAL EXCITATION

The program execution controls must also be determined by the user and are:

FINTIM ----- the maximum simulation value for the independent variable, time

DELT ----- the simulation interval; the unit of time for the integration routine to accomplish an integration step

INTEG RKSFX - the fixed integration routine with an integration interval equal to one-half of the simulation interval, DELT

FINTIM must be expressed numerically in DSL/360 and, in this program, is set arbitrarily at 10,000 seconds to ensure sufficient time for the response to attain the steady-state condition. Once steady-state is attained and the output peak magnitude and phase are calculated, the run is automatically terminated by a CALL ENDRUN statement.

In this program, the execution control DELT is directly related to OMEGA, the input frequency. Since DSL/360 requires that DELT also be expressed as a numerical value, the DSL/360 user must calculate the value of DELT from the equation

$$\text{DELT} = \frac{1}{N} \times \text{PERIOD} = \frac{2 \times \text{PI}}{N \times \text{OMEGA}} \text{ seconds.}$$

The first iteration calculation is made at time zero, and an iteration calculation is made at discrete intervals thereafter, the length of the intervals depending on the integration scheme being used [Ref. 1].

Since the output phase is not known in advance, assurance that an iteration calculation will be made at or reasonably near the peak-value point of each output cycle is made by setting a high value for N. Since the basis for steady-state testing is the comparison of values at each iteration point on one output cycle with the values at the corresponding point on two other output cycles, N should be an integer.

It has been determined empirically that the value of N should be an

integer no less than about twenty; this figure provides a trade-off between acceptable accuracy and reasonable computation time. A program for generating a table of DELT versus OMEGA, for N equal to twenty, is shown in Appendix B.

#### C. THE INITIAL REGION

The INITIAL REGION encompasses calculations, input and output operations, and initializations that must be made once only at the beginning of a run. The values and equations shown in the sample program apply to all systems and need not be changed by the user.

All variables shown in the INITIAL REGION in the sample program are either self-explanatory or explained more appropriately elsewhere in this chapter.

#### D. THE DERIVATIVE REGION

The DERIVATIVE REGION encompasses those calculations involving integration and derivatives of the state variables being integrated. The basic interval in the independent variable, time, for each pass through this region is the calculation interval. In this program, the interval is determined by INTEG RKSFX and is one-half the simulation interval, DELT.

In the program, the DERIVATIVE REGION contains equations describing the input and system transfer functions and input-output relationships. While the DSL/360 function block TRNFR is used in the sample program, representing the system in either its closed-loop or open-loop form, the system could also have been represented by a set of ordinary differential equations [Ref. 1]. The use of ordinary differential equations does not alter the remainder of the program in any way.

STORAG, a DSL/360 translator command, and TABLE, a DSL/360 data input, are both associated with the use of the function block TRNFR. SINE() is the DSL/360 function block that models the input.

#### E. THE DYNAMIC REGION

The DYNAMIC REGION encompasses all time-dependent algebraic calculations, other than derivative calculations for the integration routines in the DERIVATIVE REGION, plus all other operations which must be performed at each discrete value of time; i.e., all those operations which must be performed each iteration. It is in this region, in the sample program, that steady-state testing and response magnitude and phase calculations are accomplished.

The variables in the DYNAMIC REGION of the sample program are:

OUT ---- the value of the output magnitude at the point on the output cycle at which the current iteration calculations are being made

DELOUT - the value of OUT, to be saved for comparison with the value of OUT at the same point on the third cycle following

DILOUT - the value of OUT, to be saved for comparison with the value of OUT at the same point on the tenth cycle following

EPS1 --- the allowable difference between DELOUT and OUT (the difference between the current value of the output magnitude and the value at the same point on the third cycle preceding) within which DELOUT and OUT are considered to be equal

MAXOUT - the maximum value of OUT; the peak value of the output magnitude

DEROUT - the value of the slope of the output at the iteration point

EPS2 --- the allowable difference between DELOUT and OUT and between DILOUT and OUT within which DELOUT, DILOUT, and OUT are considered to be equal

EPS3 --- the maximum allowable value within which the slope of the output, DEROUT, may be considered to be at the peak of an output cycle

AXIN --- the time at which the input cycle crosses its zero-value point

XIN ---- the value of AXIN saved until the input cycle next crosses its zero-value point

AXOUT -- the time at which the output cycle crosses its zero-value point

XOUT --- the value of AXOUT saved until the output cycle next crosses its zero-value point

XTDIFF - the time elapsed between XIN and XOUT

RPHASE - the value of the output phase in radians

DPHASE - the value of the output phase in degrees

ARG ---- the value of the argument for the calculation of the output slope, DEROUT

ERG ---- the value of the argument for the calculation of EPS3

EPS1 is relatively coarse and is the criterion to be met by the output magnitude before the program will permit steady-state testing during any given iteration. EPS2 is a refined criterion for determining whether or not the train of cycles is sufficiently similar for the output to be considered in the steady-state condition. The run is continued, however, even after the EPS2 criterion is satisfied, until the peak value of the output magnitude has been attained, determined, in part, by EPS3.

The numbers 605 and 2005 appearing in the arguments of the equations for DELOUT and DILOUT represent the maximum number of sampled values of OUT stored in the delay intervals DELINT and DILINT, respectively. Reference 1 states that these numbers, which must be coded explicitly as numerical, integer constants, should be less than or equal to the delay interval divided by DELT. In practice, however, it was found that these numbers must be greater than the delay interval divided by DELT. For an N of twenty, then, in the case of DELOUT, this number is

$$\frac{\text{DELINT}}{\text{DELT}} > \frac{3 \times \text{PERIOD}}{1/20 \times \text{PERIOD}} = 60.$$

For DILOUT, the number is

$$\frac{DILINT}{DELTA} > \frac{10 \times PERIOD}{1/20 \times PERIOD} = 200.$$

It can be seen, therefore, that the sample program will handle a program for any DELT where N is 204 or less. If more than this number of iterations per cycle is desired, the integer numbers to replace the numbers 605 and 2005 may be determined from the above equations.

ERG is the argument, in radians, for the slope of the output at the point on the cycle most remote from the maximum point which can possibly be obtained for the DELT specified. Ideally, an iteration calculation will occur at the maximum, and the slope will be zero. However, because the process is an iterative one and the phase is shifted by an indeterminable amount, an iteration calculation may occur as far away, in time, as one-half of DELT; i.e., the maximum point may be exactly half-way between two iteration calculation points. To ensure that the slope criterion, EPS3, will be no greater than the slope at this point, ERG is determined as follows:

$$\begin{aligned} \text{Nr. of radians/iteration} &= \frac{2 \times \pi}{\text{Nr. of iterations/cycle}} \\ &= \frac{2 \times \pi}{PERIOD/DELTA} \\ &= \frac{2 \times \pi \times DELTA}{PERIOD} \end{aligned}$$

$$\begin{aligned} ERG &= \frac{\pi}{2} - \frac{1}{2} \left[ \frac{2 \times \pi \times DELTA}{PERIOD} \right] \\ &= \pi \left[ 0.5 - \frac{DELTA}{PERIOD} \right] \end{aligned}$$

Phase is calculated in the program so as to produce the result always as a lag indication by XTDIFF, else both the true phase angle and its supplement would be calculated. For a phase angle from zero degrees to 180 degrees, the program is satisfactory. However, because

all attempts to limit phase computations only to times at which the input and output cycles crossed their respective zero-value points in an upward (increasing value) direction failed, phase lag angles from 180 degrees to 360 degrees (phase lead) are calculated as supplements of the true phase angle. To determine whether the program has calculated true phase or its supplement, the user may either sketch the asymptotes of the phase plot on a Bode diagram or, if the system description does not readily lend itself to this method, have the program provide a time graph of a few cycles each of the input and the output, from which phase lag or lead may be determined visually.

#### F. THE TERMINAL REGION

Entry into the TERMINAL REGION is made at the termination of a run for purposes of performing input and output operations, testing terminating conditions, processing results, changing parameter values and requesting a rerun, or terminating the job. In the sample program, the values of the variables and parameters desired in the printout are caused to be written in the TERMINAL REGION.

If it is desired to make several runs in one job, each run at a different value of OMEGA, say, it is within this region that new values for the parameter OMEGA and the simulation interval DELT may be specified. Appendix A contains an example of such a multi-run job.

A method for programming a multi-run job with different values for OMEGA for each run, calculated by the program, and for which DELT need not be changed for each run is discussed in Chapter IV.



#### IV. CONCLUSIONS

This chapter presents a discussion of the results obtained, several limitations of the program as evolved, and recommendations for extensions of investigation.

##### A. RESULTS

Overall, the results of testing several linear systems were satisfactory. The response data provided by the program were consistently accurate, signifying that the steady-state testing scheme worked satisfactorily and consistently avoided chance satisfaction of the steady-state criteria during the settling phase.

Programs and results for several of the systems tested are shown in Appendix A.

##### B. LIMITATIONS

There are several limitations on the use of the program evolved, some inherent in DSL/360 and some a function of the value of parameters selected.

Limitation 1. DELT must be calculated by the user and expressed in the program numerically. This could be eliminated if DELT could be expressed as a variable function of OMEGA, but this requires an alteration of DSL/360, beyond the scope of this paper.

Basically, a new value for DELT must be calculated for each value of OMEGA used. However, the number of calculations for values of DELT may be reduced in a multi-run job by starting with a high value for OMEGA and a corresponding value for DELT (equal to, say, one-twentieth

of the period) and including in the program a scheme to halve the preceding value of OMEGA each run. DELT need not be changed, then, since each time OMEGA is halved, the number of iterations per cycle is doubled, and the criteria for DELT are satisfied (N remains an integer no smaller than twenty, doubling each successive run; the program as shown in the sample will accept any value of N to a maximum of about 204). Appendix A contains an example of this method.

Limitation 2. Since the independent variable is time, only time plots may be obtained without modification of DSL/360. Such plots may be desired if the user wants to determine whether phase lags or leads.

Limitation 3. Use of too small a value for OMEGA, as compared to the value for the natural frequency of the system, may result in no program output. This is caused by the fixed integration called for by INTEG RKSFX. If this card is removed from the deck, the integration interval automatically defaults to that called for by the variable-step integration routine RKS. However, DELT then varies with the integration interval [Ref. 1], and the conditions required for steady-state testing may not be met because integration points on different cycles may not correspond. It was attempted to begin a run with the variable-step routine RKS and, after some relatively short interval, to impose the fixed-step routine RKSFX. These attempts were unsuccessful, however, because no matter where the INTEG RKSFX card was located -- even within a NOSORT region [Ref. 1] -- it was employed from the outset.

#### C. RECOMMENDATIONS FOR EXTENSIONS OF INVESTIGATION

There are several recommendations for logical extensions of this investigation.

Recommendation 1. Determine a method for calculating phase that will not result in computing the supplement of the phase angle, rather than the true phase angle, whenever the phase is leading (or, as written in the program, whenever the phase lags from 180 degrees to 360 degrees).

Recommendation 2. Alter DSL/360 to permit the designation of DELT as a variable function of OMEGA rather than as a numerical constant.

Recommendation 3. Alter DSL/360 to permit graphing a variable versus some other parameter than time; specifically, it would be desirable to graph both MAXOUT and the phase versus OMEGA.

Recommendation 4. Apply the basic concept of this paper to the measurement of frequency response for a non-linear system.

## APPENDIX A

This appendix presents six examples of systems used to test the program evolved for determining the frequency response. For each example, the system description, the program, and the frequency response is given.

### EXAMPLE 1

The system used in this example is identical to the system used in the sample program of Figure 3.1. This example shows one method of obtaining the response to more than one frequency in one job. DELT must be calculated and specified within the program for each OMEGA. The program is shown in Figure A.1, and the frequency response is shown in Figure A.2.

For this example, the simulation time varied from 3.65 seconds to 8.11 seconds for runs for nine different frequencies, or an average of 6.05 seconds per run. The total execution time was 58.49 seconds.

```

TITLE    LINEAR SYSTEM RESPONSE TO A SINE WAVE INPUT
*        FORWARD-PATH TRANSFER FUNCTION: 100/S(S+1)
*        FEEDBACK-PATH TRANSFER FUNCTION: -1
CONST PI=3.1415927
PARAM OMEGA=6.,A=1.,FGAIN=1.
CONTRL FINTIM=10000.,DELT=0.05235986
INTEG RKSF
INITIAL REGION
PERIOD=2.*PI/OMEGA
DELINT=3.*PERIOD
DILINT=10.*PERIOD
ERG=PI*(.5-DELT/PERIOD)
EPS1=1.E-03*A
EPS2=0.
EPS3=0.
XIN=0.
XOUT=0.
XTDIFF=0.
RPHASE=0.
DPHASE=0.
MAXOUT=0.
DEROUT=10.
DERIVATIVE REGION
IN=A*SINE(0.,OMEGA,0.)
ERROR=IN-FGAIN*OUT
OUT=TRNFR(0.,2.,IC,NUM,DEN,ERROR)
STORAG IC(2),NUM(1),DEN(3)
TABLE IC(1-2)=0.,0.,NUM(1)=100.,DEN(1-3)=1.,1.,0.
DYNAMIC REGION
* DETERMINATION OF STEADY-STATE CONDITION
DELOUT=DELAY(605,DELINT,OUT)
DILOUT=DELAY(2005,DILINT,OUT)
IF(TIME.LE.DELINT) GO TO 2
IF(ABS(DELOUT-OUT).LE.EPS1) GO TO 1
MAXOUT=0.
GO TO 2
* DETERMINATION OF OUTPUT STEADY-STATE PEAK MAGNITUDE
1 IF(OUT.GT.MAXOUT) MAXOUT=OUT
DEROUT=OMEGA*MAXOUT*CCS(ERG)
EPS2=0.001*MAXOUT
EPS3=OMEGA*MAXOUT*CCS(ERG)
* DETERMINATION OF OUTPUT STEADY-STATE PHASE
2 AXIN=CROSS(TIME,IN,0.)
IF(AXIN.NE.0.) XIN=AXIN
AXOUT=CFOSS(TIME,OUT,0.)
IF(AXOUT.NE.0.) XOUT=AXOUT
IF(XIN.GT.XOUT) GO TO 3
XTDIFF=XIN-XOUT
RPHASE=2*PI*XTDIFF/PERIOD
DPHASE=360.*XTDIFF/PERIOD
3 ARG=OMEGA*TIME+RPHASE
IF(TIME.LE.DILINT) GO TO 4
IF(ABS(DELOUT-OUT).LE.EPS2.AND.ABS(DILOUT-OUT...
).LE.EPS2.AND.ABS(DEROUT...
).LE.EPS3.AND.OUT.GT.0..AND.MAXOUT.GT.0.) CALL ENDRUN
4 CONTINUE
TERMINAL REGION
PRINT OMEGA,MAXOUT,RPHASE,DPHASE
CALL PRINT
END
PARAM OMEGA=7.
CONTRL DELT=0.04487989
END

```

FIGURE A.1  
PROGRAM FOR SYSTEM OF EXAMPLE 1

```
PARAM OMEGA=8.  
CONTRL DELT=0.03926991  
END  
PARAM OMEGA=9.  
CONTRL DELT=0.03490659  
END  
PARAM OMEGA=10.  
CONTRL DELT=0.031415927  
END  
PARAM OMEGA=11.  
CONTRL DELT=0.02855993  
END  
PARAM OMEGA=12.  
CONTRL DELT=0.02617994  
END  
PARAM OMEGA=13.  
CONTRL DELT=0.02416609  
END  
PARAM OMEGA=14.  
CONTRL DELT=0.02243995  
END  
STOP
```

FIGURE A.1 (CONTINUED)

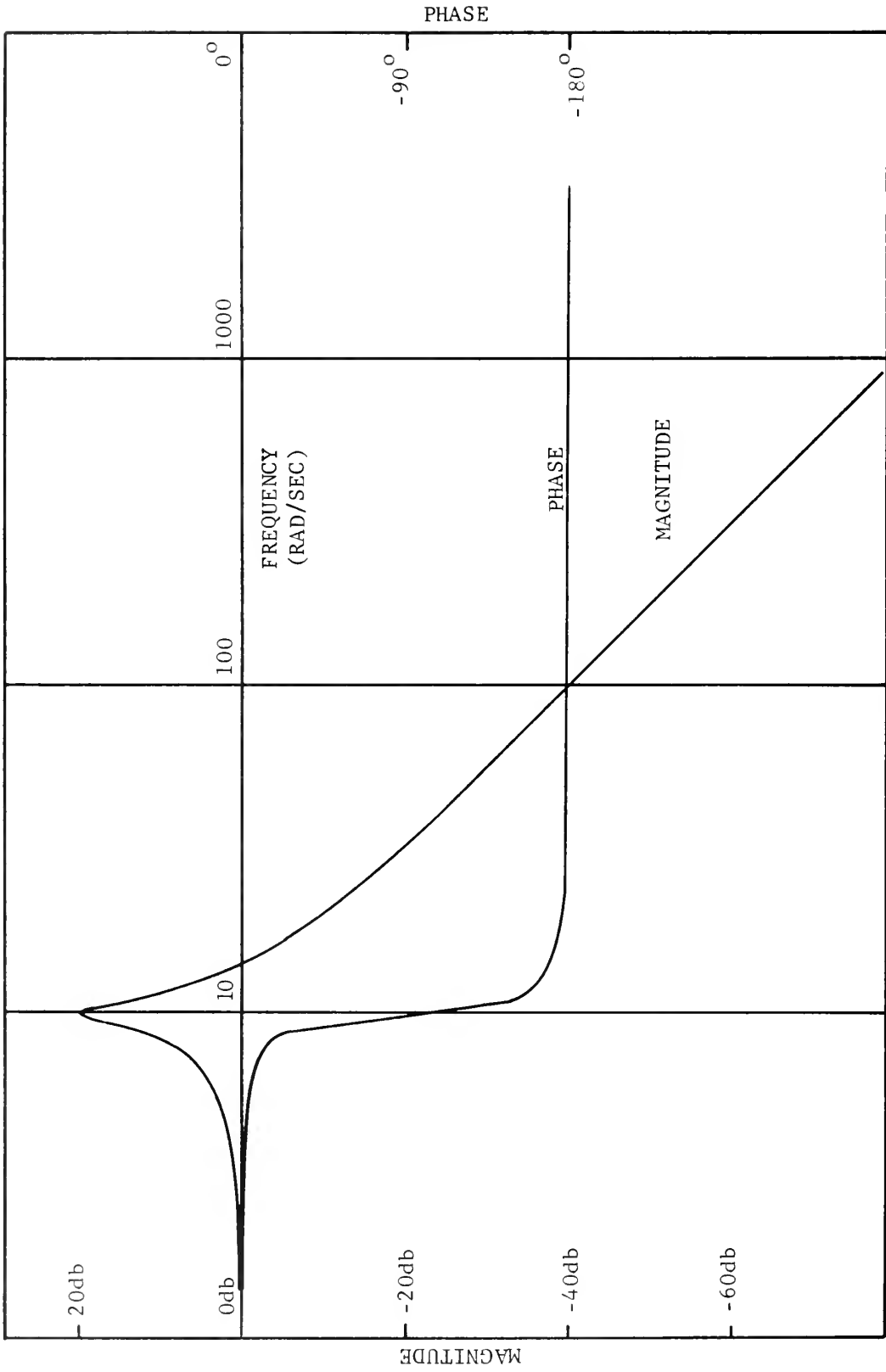


FIGURE A.2

Frequency Response for System of Examples 1 - 3



## EXAMPLE 2

The system used in this example is identical to the system used in the sample program of Figure 3.1. This example shows a second method, a variation of that shown in Example 1, whereby the response to more than one frequency may be obtained in one job. For this method, DELT need not be recalculated for each OMEGA, which is halved each run, since halving OMEGA results in doubling the number of iteration intervals per cycle for a given DELT. To ensure that the criteria for DELT (and N) are always met, the highest value for OMEGA for which a response is desired in the job is specified in the PARAM card.

The program is shown in Figure A.3. The frequency response, identical to that obtained for Example 1, is shown in Figure A.2.

Since the number of iterations per cycle doubles each run, simulation time for this method is greater than for the method of Example 1, where the number of iterations per cycle remains constant for all frequencies (so long as the same value for N is used). For a maximum allowable simulation time of ten minutes (a local ruling specified for the computer on which these examples were run), the responses for a maximum of only six frequencies were obtainable on any single job, an average of 100 seconds per run. The method of Example 1 is, therefore, in general less costly. In general, simulation time may vary considerably from one system to another and, for any given system, from one frequency to another.

```

TITLE    LINEAR SYSTEM RESPONSE TO A SINE WAVE INPUT
*        FORWARD-PATH TRANSFER FUNCTION: 100/S(S+1)
*        FEEDBACK-PATH TRANSFER FUNCTION: -1
CONST PI=3.1415927
PARAM OMEGA=40., LOMEGA=1., A=1., FGAIN=1.
CONTRL FINTIM=10000., DELT=0.015707964
INTEG RKSFX
INITIAL REGION
    OMEGA=OMEGA/2.
    PERIOD=2.*PI/OMEGA
    DELINT=3.*PERIOD
    DILINT=10.*PERIOD
    ERG=PI*(.5-DELT/PERIOD)
    EPS1=1.E-03*A
    EPS2=0.
    EPS3=0.
    XIN=0.
    XOUT=0.
    XTDIFF=0.
    RPHASE=0.
    DPHASE=0.
    MAXOUT=0.
    DEROUT=10.
DERIVATIVE REGION
    IN=A*SINE(0., OMEGA, 0.)
    ERROR=IN-FGAIN*OUT
    OUT=TRNFR(0., 2., IC, NUM, DEN, ERROR)
STORAG IC(2), NUM(1), DEN(3)
TABLE IC(1-2)=C., 0., NUM(1)=100., DEN(1-3)=1., 1., 0.
DYNAMIC REGION
* DETERMINATION OF STEADY-STATE CONDITION
    DELOUT=DELAY(605, DELINT, OUT)
    DILOUT=DELAY(2005, DILINT, OUT)
    IF(TIME.LE.DELINT) GO TO 2
    IF(ABS(DELOUT-OUT).LE.EPS1) GO TO 1
    MAXOUT=0.
    GO TO 2
* DETERMINATION OF OUTPUT STEADY-STATE PEAK MAGNITUDE
    1 IF(OUT.GT.MAXOUT) MAXOUT=OUT
    DEROUT=OMEGA*MAXOUT*COS(ARG)
    EPS2=0.001*MAXOUT
    EPS3=OMEGA*MAXOUT*COS(ERG)
* DETERMINATION OF OUTPUT STEADY-STATE PHASE
    2 AXIN=CROSS(TIME, IN, 0.)
    IF(AXIN.NE.0.) XIN=AXIN
    AXOUT=CROSS(TIME, OUT, 0.)
    IF(AXOUT.NE.0.) XOUT=AXOUT
    IF(XIN.GT.XOUT) GO TO 3
    XTDIFF=XIN-XOUT
    RPHASE=2*PI*XTDIFF/PERIOD
    DPHASE=360.*XTDIFF/PERIOD
    3 ARG=OMEGA*TIME+RPHASE
    IF(TIME.LE.DILINT) GO TO 4
    IF(ABS(DELOUT-OUT).LE.EPS2.AND.ABS(DILOUT-OUT...
    ).LE.EPS2.AND.ABS(DEROUT...
    ).LE.EPS3.AND.OUT.GT.0..AND.MAXOUT.GT.0.) CALL ENDRUN
    4 CONTINUE
TERMINAL REGION
PRINT OMEGA, MAXOUT, RPHASE, DPHASE
CALL PRINT
IF(OMEGA.GE.LOMEGA) CALL RERUN
END
STOP

```

FIGURE A.3  
PROGRAM FOR SYSTEM OF EXAMPLE 2

### EXAMPLE 3

This example, too, uses the system shown in the sample program of Figure 3.1 and in Examples 1 and 2. In this case, however, the system is modeled in its open-loop form, rather than the closed-loop form employed in the sample and Examples 1 and 2. The program is shown in Figure A.4, and its frequency response, identical to those obtained in Examples 1 and 2, is shown in Figure A.2.

```

TITLE    LINEAR SYSTEM RESPONSE TO A SINE WAVE INPUT
*        FORWARD-PATH TRANSFER FUNCTION: 100/(S2+S+100)
*        FEEDBACK-PATH TRANSFER FUNCTION: ZERO
CONST   PI=3.1415927
PARAM   OMEGA=160.,LOMEGA=1.,A=1.
CONTRL  FINTIM=10000.,DELT=0.003926989
INTEG   RKSFX
INITIAL REGION
        OMEGA=OMEGA/2.
        PERIOD=2.*PI/OMEGA
        DELINT=3.*PERIOD
        DILINT=10.*PERIOD
        ERG=PI*(.5-DELT/PERIOD)
        EPS1=1.E-03*A
        EPS2=0.
        EPS3=0.
        XIN=0.
        XOUT=0.
        XTDIFF=0.
        RPHASE=0.
        DPHASE=0.
        MAXOUT=0.
        DEROUT=10.
DERIVATIVE REGION
        IN=A*SINE(0.,OMEGA,0.)
        OUT=TRNFR(0.,2.,IC,NUM,DEN,IN)
STORAG  IC(2),NUM(1),DEN(3)
TABLE   IC(1-2)=0.,0.,NUM(1)=100.,DEN(1-3)=1.,1.,100.
DYNAMIC REGION
*   DETERMINATION OF STEADY-STATE CONDITION
        DELOUT=DELAY(605,DELINT,OUT)
        DILOUT=DELAY(2005,DILINT,OUT)
        IF(TIME.LE.DELINT) GO TO 2
        IF(ABS(DELOUT-OUT).LE.EPS1) GO TO 1
        MAXOUT=0.
        GO TO 2
*   DETERMINATION OF OUTPUT STEADY-STATE PEAK MAGNITUDE
        1 IF(OUT.GT.MAXOUT) MAXOUT=OUT
        DEROUT=OMEGA*MAXOUT*COS(ERG)
        EPS2=0.001*MAXOUT
        EPS3=OMEGA*MAXOUT*COS(ERG)
*   DETERMINATION OF OUTPUT STEADY-STATE PHASE
        2 AXIN=CROSS(TIME,IN,0.)
        IF(AXIN.NE.0.) XIN=AXIN
        AXOUT=CROSS(TIME,OUT,0.)
        IF(AXOUT.NE.0.) XOUT=AXOUT
        IF(XIN.GT.XOUT) GO TO 3
        XTDIFF=XIN-XOUT
        RPHASE=2*PI*XTDIFF/PERIOD
        DPHASE=360.*XTDIFF/PERIOD
        3 ARG=OMEGA*TIME+RPHASE
        IF(TIME.LE.DILINT) GO TO 4
        IF(ABS(DELOUT-OUT).LE.EPS2.AND.ABS(DILOUT-OUT...
        ).LE.EPS2.AND.ABS(DEROUT...
        ).LE.EPS3.AND.OUT.GT.0..AND.MAXOUT.GT.0.) CALL ENDRUN
        4 CONTINUE
TERMINAL REGION
PRINT   OMEGA,MAXOUT,RPHASE,DPHASE
        CALL PRINT
        IF(OMEGA.GE.LOMEGA) CALL RERUN
END
STOP

```

FIGURE A.4  
PROGRAM FOR SYSTEM OF EXAMPLE 3

#### EXAMPLE 4

The system used in this example is open-loop, with a transfer function of  $10/s(s+1)(s+5)$ . The program is shown in Figure A.5, and the system response is shown in Figure A.6.

```

TITLE    LINEAR SYSTEM RESPONSE TO A SINE WAVE INPUT
*        FORWARD-PATH TRANSFER FUNCTION:  10/(S3+6S2+5S-10)
*        FEEDBACK-PATH TRANSFER FUNCTION: -1
CONST   PI=3.1415927
PARAM   OMEGA=200.,LOMEGA=0.01,A=1.
CONTRL  FINTIM=10000.,DELT=0.0015707964
INTEG   RKSFX
INITIAL REGION
        OMEGA=OMEGA/2.
        PERIOD=2.*PI/OMEGA
        DELINT=3.*PERIOD
        DILINT=10.*PERIOD
        ERG=PI*(.5-DELT/PERIOD)
        EPS1=1.E-03*A
        EPS2=0.
        EPS3=0.
        XIN=0.
        XOUT=0.
        XTDIFF=0.
        RPHASE=0.
        DPHASE=C.
        MAXOUT=0.
        DEROUT=10.
DERIVATIVE REGION
        IN=A*SINE(0.,OMEGA,0.)
        OUT=TRNFR(0.,3.,IC,NUM,DEN,IN)
STORAG  IC(3),NUM(1),DEN(4)
TABLE   IC(1-3)=0.,0.,0.,NUM(1)=10.,DEN(1-4)=1.,6.,5.,0.
DYNAMIC REGION
* DETERMINATION OF STEADY-STATE CONDITION
        DELOUT=DELAY(605,DELINT,OUT)
        DILOUT=DELAY(2005,DILINT,OUT)
        IF(TIME.LE.DELINT) GO TO 2
        IF(ABS(DELOUT-OUT).LE.EPS1) GO TO 1
        MAXOUT=0.
        GO TO 2
* DETERMINATION OF OUTPUT STEADY-STATE PEAK MAGNITUDE
        1 IF(OUT.GT.MAXOUT) MAXOUT=OUT
          DEPOUT=OMEGA*MAXOUT*COS(ARG)
          EPS2=0.001*MAXOUT
          EPS3=OMEGA*MAXOUT*COS(ERG)
* DETERMINATION OF OUTPUT STEADY-STATE PHASE
        2 AXIN=CROSS(TIME,IN,0.)
          IF(AXIN.NE.0.) XIN=AXIN
          AXOUT=CROSS(TIME,OUT,0.)
          IF(AXOUT.NE.0.) XOUT=AXOUT
          IF(XIN.GT.XOUT) GO TO 3
          XTDIFF=XIN-XOUT
          RPHASE=2*PI*XTDIFF/PERIOD
          DPHASE=360.*XTDIFF/PERIOD
        3 ARG=OMEGA*TIME+RPHASE
          IF(TIME.LE.DILINT) GO TO 4
          IF(ABS(DELOUT-OUT).LE.EPS2.AND.ABS(DILOUT-OUT...
          ).LE.EPS2.AND.ABS(DEROUT...
          ).LE.EPS3.AND.OUT.GT.0..AND.MAXOUT.GT.0.) CALL ENDRUN
        4 CONTINUE
TERMINAL REGION
PRINT   OMEGA,MAXOUT,RPHASE,DPHASE
        CALL PRINT
        IF(OMEGA.GE.LOMEGA) CALL RERUN
END
STOP

```

FIGURE A.5  
PROGRAM FOR SYSTEM OF EXAMPLE 4

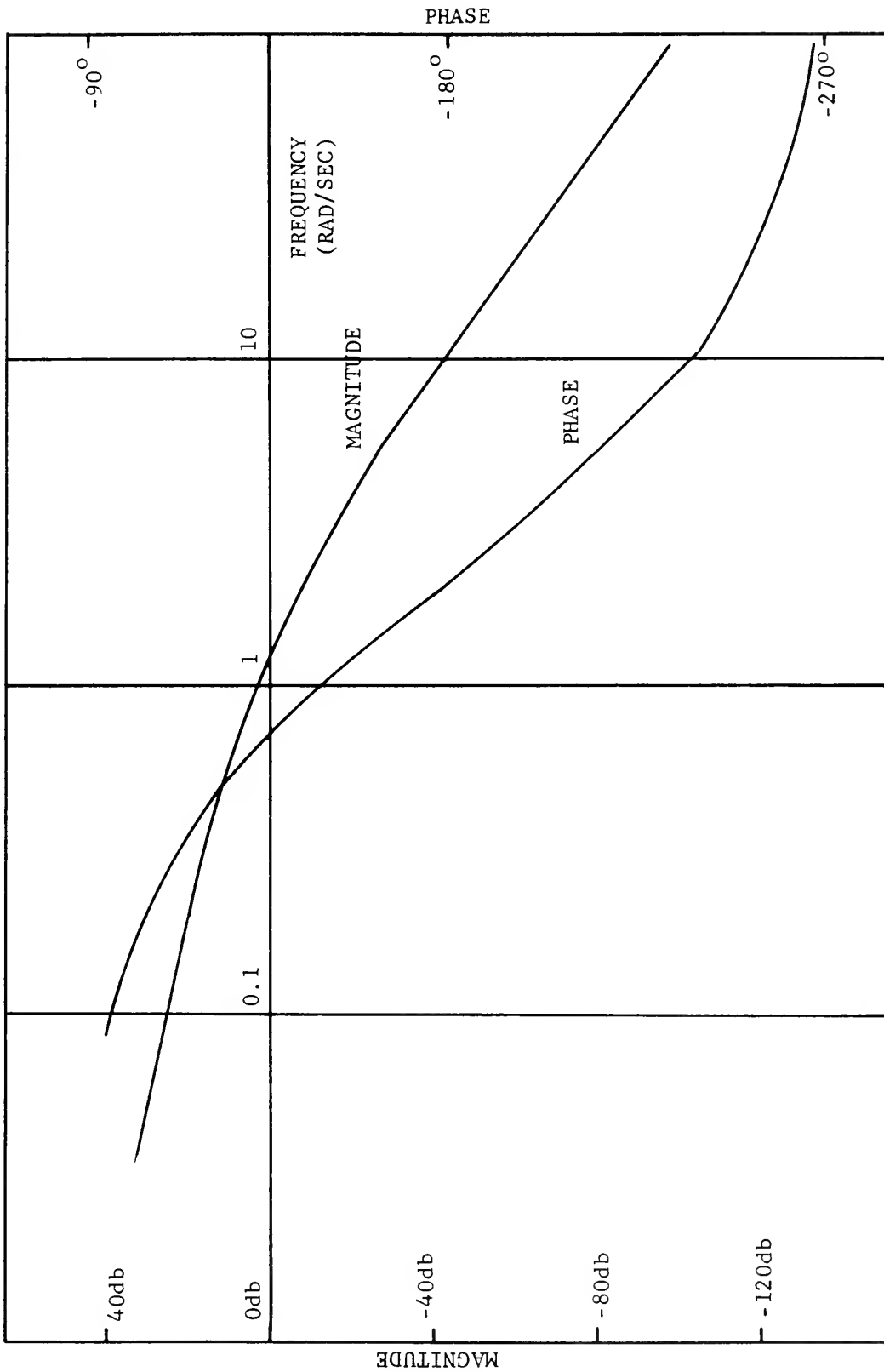


FIGURE A.6

Frequency Response for System of Example 4

### EXAMPLE 5

The system used in this example is open-loop, with a transfer function of  $(s+1)/(s+10)$ . To meet DSL/360 criteria for system modeling [Ref. 1], the division was carried out, yielding  $1 - 9/(s+10)$ . The system was then modeled from the diagram shown in Figure A.7.

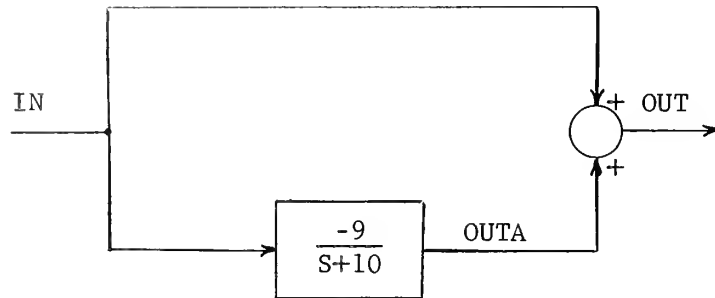


FIGURE A.7

#### Block Diagram of System for Example 5

The program is shown in Figure A.8, and the frequency response is shown in Figure A.9.

The phase of the output in this system leads the input. One limitation to the program, discussed in Chapter IV, is that the program computes the supplement of the phase angle rather than the true value of the phase whenever the output leads the input, as in this example. True phase is used in the plot of Figure A.9.



```

TITLE    LINEAR SYSTEM RESPONSE TO A SINE WAVE INPUT
*        FORWARD-PATH TRANSFER FUNCTION:  (S+1)/(S+10) =
*        1 - 9/(S+10)
*        FEEDBACK-PATH TRANSFER FUNCTION: ZERO
CONST PI=3.1415927
PARAM OMEGA=160.,LOMEGA=1.,A=1.
CONTRL FINTIM=10000.,DELT=0.003926989
INTEG RKSF
INITIAL REGION
OMEGA=OMEGA/2.
PERIOD=2.*PI/OMEGA
DELINT=3.*PERIOD
DILINT=10.*PERIOD
ERG=PI*(.5-DELT/PERIOD)
EPS1=1.E-03*A
EPS2=0.
EPS3=0.
XIN=0.
XOUT=0.
XTDIFF=0.
RPHASE=0.
DPHASE=0.
MAXOUT=0.
DEROUT=10.
DERIVATIVE REGION
IN=A*SINE(0.,OMEGA,0.)
OUTA=TRNFR(0.,1.,IC,NUM,DEN,IN)
STORAG IC(1),NUM(1),DEN(2)
TABLE IC(1)=0.,NUM(1)=-9.,DEN(1-2)=1.,10.
OUT=OUTA+IN
DYNAMIC REGION
* DETERMINATION OF STEADY-STATE CONDITION
DELOUT=DELAY(605,DELINT,OUT)
DILOUT=DELAY(2005,DILINT,OUT)
IF(TIME.LE.DELINT) GO TO 2
IF(ABS(DELOUT-OUT).LE.EPS1) GO TO 1
MAXOUT=0.
GO TO 2
* DETERMINATION OF OUTPUT STEADY-STATE PEAK MAGNITUDE
1 IF(OUT.GT.MAXOUT) MAXOUT=OUT
DEROUT=OMEGA*MAXOUT*CGS(ARG)
EPS2=0.001*MAXOUT
EPS3=OMEGA*MAXOUT*COS(ERG)
* DETERMINATION OF OUTPUT STEADY-STATE PHASE
2 AXIN=CROSS(TIME,IN,0.)
IF(AXIN.NE.0.) XIN=AXIN
AXOUT=CROSS(TIME,OUT,0.)
IF(AXOUT.NE.0.) XOUT=AXOUT
IF(XIN.GT.XOUT) GO TO 3
XTDIFF=XIN-XOUT
RPHASE=2*PI*XTDIFF/PERIOD
DPHASE=360.*XTDIFF/PERIOD
3 ARG=OMEGA*TIME+RPHASE
IF(TIME.LE.DILINT) GO TO 4
IF(ABS(DELOUT-OUT).LE.EPS2.AND.ABS(DILOUT-OUT...
).LE.EPS2.AND.ABS(DEROUT...
).LE.EPS3.AND.OUT.GT.0..AND.MAXOUT.GT.0.) CALL ENDRUN
4 CONTINUE
TERMINAL REGION
PRINT OMEGA,MAXOUT,RPHASE,DPHASE
CALL PRINT
IF(OMEGA.GE.LOMEGA) CALL RERUN
END
STOP

```

FIGURE A.8  
PROGRAM FOR SYSTEM OF EXAMPLE 5

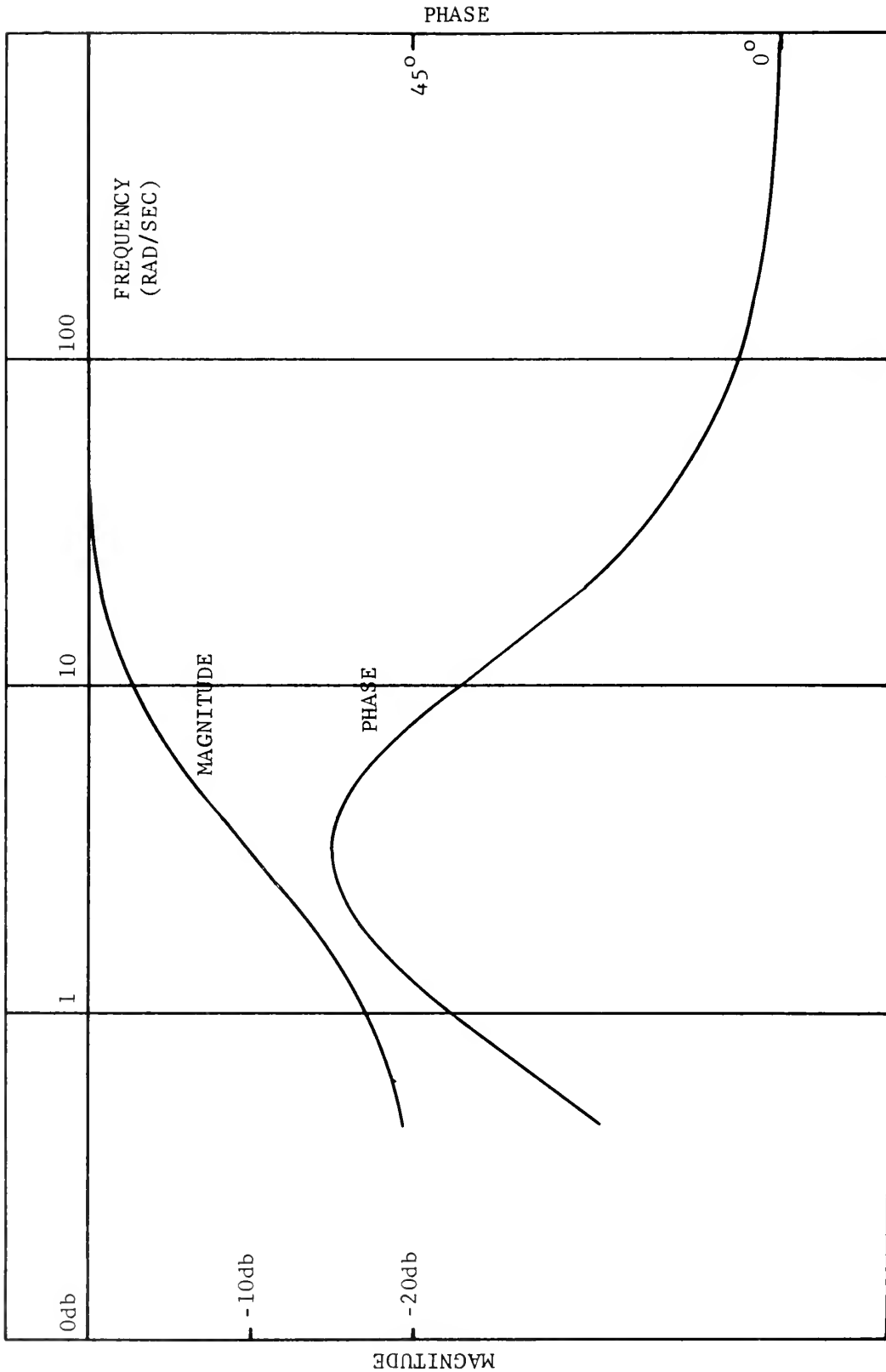


FIGURE A.9

Frequency Response for System of Example 5

## EXAMPLE 6

The system modeled in this example is a magnetic tape transport mechanism which employs a vacuum column for tape loop control. The equations representing the system have been linearized for use with the program.

The input is an AC signal to a DC motor, and the output is the velocity of the tape before it passes over the capstan.

A diagram of the system is shown in Figure A.10, and the program containing the system modeling equations is shown in Figure A.11. The frequency response of the system is shown in Figure A.12.

It will be noted that a resonance peak occurs at a frequency of approximately 20,000 radians per second. All attempts to determine frequency response at or near this frequency were unsuccessful because the steady-state criteria were not met within ten minutes, the maximum simulation time available locally on the computer on which these examples were run.

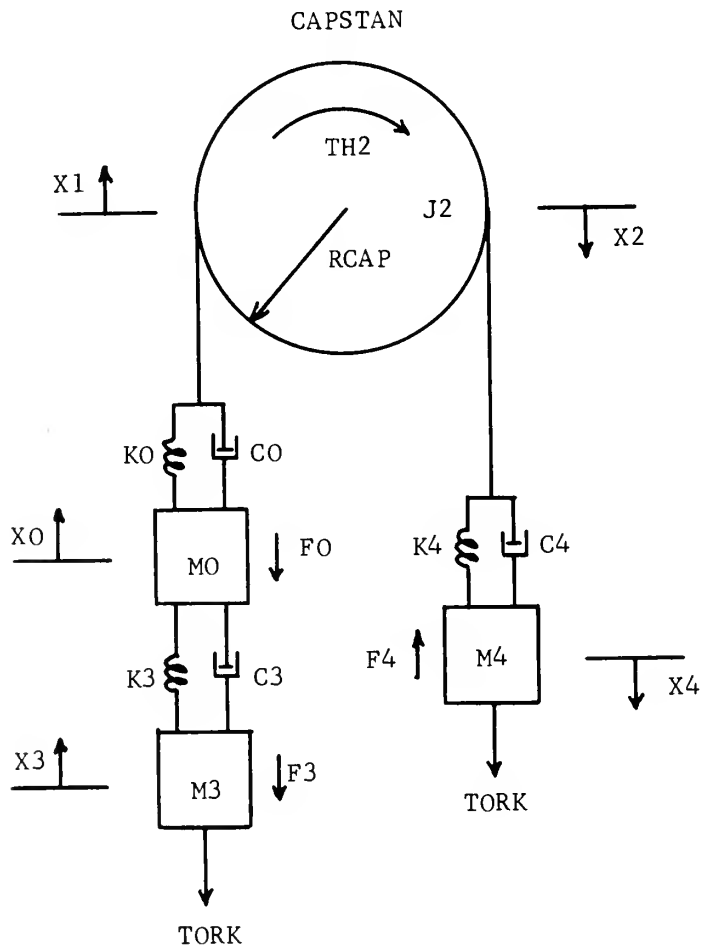
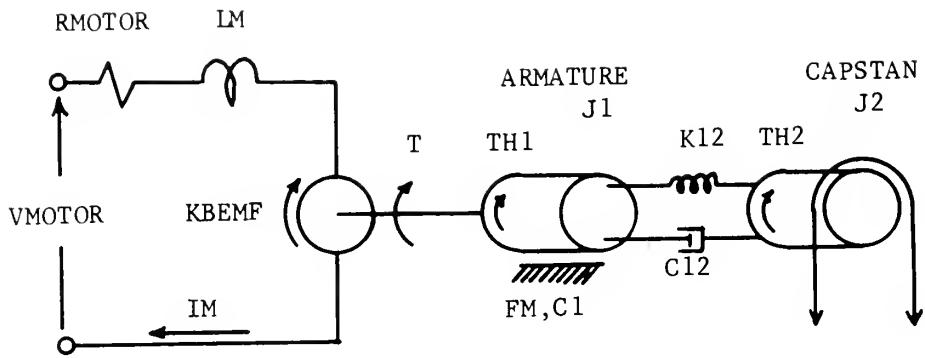


FIGURE A.10

Diagram of System for Example 6

```

TITLE    LINEAR SYSTEM RESPONSE TO A SINE WAVE INPUT
*CONST  SYSTEM--LINEARIZED CYBERNET CAPSTAN MOTOR,TAPE,AN
CONST   PI=3.1415927,....
        M4=19.245E-06,K4=119.8,C4=8.5E-03,RCAP=.75,....
        LM=20.0E-06,RMOTOR=0.6,KBEMF=.0397,MO=2.115E-06,....
        K3=135.,C3=.0092,KO=1091.,CO=.0255,M3=14.15E-06,....
        J2=22.5E-06,K12=6000.,C12=0.008,J1=50.E-06,....
        KT=0.353,C1=14.F-06,....
        FM=C.0,FO=0.0,F3=0.0,F4=0.0,TORK=0.0
INCON   ICM=C.,IC01=0.,IC02=1.E-30,IC11=0.,IC12=1.E-30,....
        IC22=C.,IC31=0.,IC32=1.E-30,IC41=0.,IC42=1.E-30,....
        IC21=0.
PARAM   OMEGA=45000.,LOMEGA=1.,A=1.
CNTRL   FINTIM=10000.,DELT=0.0000139626
INTEG   RKSFX
INITIAL REGION
        OMEGA=OMEGA/2.
        PERIOD=2.*PI/OMEGA
        DELINT=3.*PERIOD
        DILINT=10.*PERIOD
        ERG=PI*(.5-DELT/PERIOD)
        EPS1=1.E-03*A
        EPS2=0.
        EPS3=0.
        XIN=C.
        XOUT=C.
        XTDIFF=0.
        RPHASE=0.
        DPHASE=C.
        MAXOUT=0.
        DEROUT=10.
DERIVATIVE REGION
        IN=A*SINE(0.,OMEGA,0.)
        VMOTOR=IN
        LMIDOT=VMOTOR-RMOTOR*IM-KBEMF*TH1DOT
        IMDOT=(1./LM)*LMIDOT
        IM=INTGRL(ICM,IMDOT)
        MOTORQ=KT*IM-C1*TH1DOT-SIGN(FM,TH1DOT)-K12*...
        (TH1-TH2)-C12*(TH1DOT-TH2DOT)
        TH1D2=(1./J1)*MOTORQ
        TH1DOT=INTGRL(IC12,TH1D2)
        TH1=INTGRL(IC11,TH1DOT)
        SUM4=TORK-T2-SIGN(F4,X4DOT)
        X4D2=(1./M4)*SUM4
        X4DOT=INTGRL(IC42,X4D2)
        X4=INTGRL(IC41,X4DOT)
        SUMX42=X4-X2
        DOTX42=X4DOT-X2DOT
        T2=C4*DOTX42+K4*SUMX42
        X2DOT=X1DOT
        X2=X1
        T12=T1-T2
        CAPTOR=K12*(TH1-TH2)+C12*(TH1DOT-TH2DOT)-RCAP*T12
        TH2D2=(1./J2)*CAPTOR
        TH2DOT=INTGRL(IC22,TH2D2)
        TH2=INTGRL(IC21,TH2DOT)
        T1=KO*SUMX01+CO*DOTX01
        SUMX01=X1-X0
        DOTX01=X1DOT-X0DOT
        X1DOT=RCAP*TH2DOT
        X1=RCAP*TH2
        SUM0=K3*X3+C3*X3DOT+K0*X1+CO*X1DOT-(CO+C3)*X0DOT-...
        (K0+K3)*X0-SIGN(FO,X0DOT)

```

FIGURE A.11  
PROGRAM FOR SYSTEM OF EXAMPLE 6

```

XOD2=(1./M0)*SUM0
XODOT=INTGRL(IC02,XOD2)
XO=INTGRL(IC01,XODOT)
SUM3=K3*(XO-X3)+C3*(XODOT-X3DOT)-TORK-SIGN(F3,X3DOT)
X3D2=(1./M3)*SUM3
X3DOT=INTGRL(IC32,X3D2)
X3=INTGRL(IC31,X3DOT)
OUT=XODOT
DYNAMIC REGION
* DETERMINATION OF STEADY-STATE CONDITION
DELOUT=DELAY(605,DELINT,OUT)
DILOUT=DELAY(2005,DILINT,OUT)
IF(TIME.LE.DELINT) GO TO 2
IF(ABS(DELOUT-OUT).LE.EPS1) GO TO 1
MAXOUT=0.
GO TO 2
* DETERMINATION OF OUTPUT STEADY-STATE PEAK MAGNITUDE
1 IF(OUT.GT.MAXOUT) MAXOUT=OUT
DEROUT=OMEGA*MAXOUT*COS(ARG)
EPS2=0.001*MAXOUT
EPS3=OMEGA*MAXOUT*COS(ERG)
* DETERMINATION OF OUTPUT STEADY-STATE PHASE
2 AXIN=CROSS(TIME,IN,0.)
IF(AXIN.NE.0.) XIN=AXIN
AXOUT=CROSS(TIME,OUT,0.)
IF(AXOUT.NE.0.) XOUT=AXOUT
IF(XIN.GT.XOUT) GO TO 3
XTDIFF=XIN-XOUT
RPHASE=2*PI*XTDIFF/PERIOD
DPHASE=360.*XTDIFF/PERIOD
3 ARG=OMEGA*TIME+RPHASE
IF(TIME.LE.DILINT) GO TO 4
IF(ABS(DELOUT-OUT).LE.EPS2.AND.ABS(DILOUT-OUT...
).LE.EPS2.AND.ABS(DEROUT...
).LE.EPS3.AND.OUT.GT.0..AND.MAXOUT.GT.0.) CALL ENDRUN
4 CONTINUE
TERMINAL REGION
PRINT OMEGA,XO,OUT,MAXOUT,X1,X1DOT,X4,X4DOT,RPHASE,DPHASE
CALL PRINT
IF(OMEGA.GE.LOMEGA) CALL RERUN
END
STOP

```

FIGURE A.11 (CONTINUED)

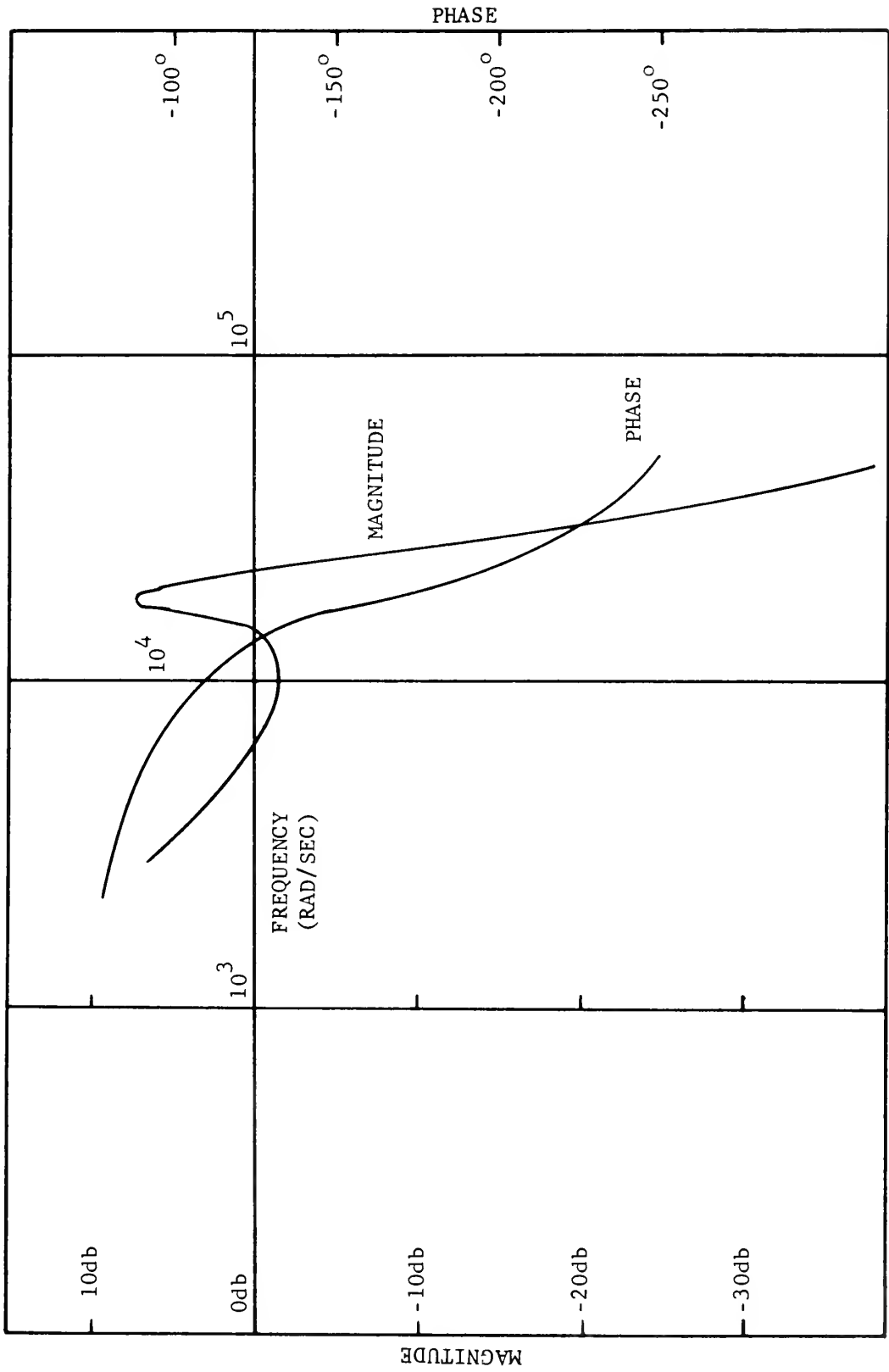


FIGURE A.12

Frequency Response for System of Example 6

## APPENDIX B

A program for generating a table of DELT versus OMEGA for an N of twenty is shown in Figure B.1. To generate a table for any other value of N, the user need only specify the desired value of N in the program. The program may also be modified to obtain values for DELT for intervals of OMEGA and a maximum value of OMEGA other than the one radian per second and the 10,000 radians per second, respectively, shown in the program.

```
C      GENERATION OF TABLE OF DELT VERSUS OMEGA FOR N=20
C
      WRITE(6,1)
1  FORMAT(1X,'TABLE OF DELT VERSUS OMEGA FOR N=20',//,1X,
      *'OMEGA',9X,'DELT',/)
      PI=3.1415927
      OMEGA=1.
      AN=20.
      DO 3 I=1,10000
      DELT=2.*PI/(AN*OMEGA)
      WRITE(6,2) OMEGA, DELT
2  FORMAT(F7.0,E18.7)
      OMEGA=OMEGA+1.
3  CONTINUE
      STOP
      END
```

FIGURE B.1

Program for Generating Table of DELT versus OMEGA



### LIST OF REFERENCES

1. International Business Machines Company Report DSL/360, Digital Simulation Language, by W. M. Syn, N. N. Turner, and D. G. Wyman, November 1968.
2. Gille, J-C, Pelegrin, M. J., and Decauline, P., Feedback Control Systems, McGraw-Hill, 1959.
3. DiStefano, J. J., Stubberud, A. R., and Williams, I. J., Feedback and Control Systems, Schaum, 1967.

INITIAL DISTRIBUTION LIST

	No. Copies
1. Defense Documentation Center Cameron Station Alexandria, Virginia 22314	20
2. Library, Code 0212 Naval Postgraduate School Monterey, California 93940	2
3. Naval Security Group Command Headquarters 3401 Nebraska Avenue Washington, D. C. 20390	1
4. Professor George J. Thaler, Code 52TR Department of Electrical Engineering Naval Postgraduate School Monterey, California 93940	5
5. CDR Richard O. Allen, Jr., USN 3401 South US 23 Greenbush, Michigan 48738	1
6. Mr. W. M. Syn International Business Machines Company Monterey and Cottle Roads (Building 014) San Jose, California 95114	1
7. Mr. Martin Dost International Business Machines Company Monterey and Cottle Roads (Building 026) San Jose, California 95114	1
8. Mr. David Turner International Business Machines Company Monterey and Cottle Roads (Building 026) San Jose, California 95114	1

## DOCUMENT CONTROL DATA - R &amp; D

(Security classification of title, body of abstract and indexing annotation must be entered when the overall report is classified)

1. ORIGINATING ACTIVITY (Corporate author)		2a. REPORT SECURITY CLASSIFICATION	
Naval Postgraduate School Monterey, California 93940		Unclassified	
		2b. GROUP	
3. REPORT TITLE			
A Digital Computer Program for Determining the Frequency Response of a Linear System to a Sinusoidal Excitation			
4. DESCRIPTIVE NOTES (Type of report and, inclusive dates)			
Master's Thesis; December 1969			
5. AUTHOR(S) (First name, middle initial, last name)			
Richard Owen Allen, Jr.			
6. REPORT DATE		7a. TOTAL NO. OF PAGES	7b. NO. OF REFS
December 1969		47	3
8a. CONTRACT OR GRANT NO.		9a. ORIGINATOR'S REPORT NUMBER(S)	
b. PROJECT NO.			
c.		9b. OTHER REPORT NO(S) (Any other numbers that may be assigned this report)	
d.			
10. DISTRIBUTION STATEMENT			
This document has been approved for public release and sale; its distribution is unlimited.			
11. SUPPLEMENTARY NOTES		12. SPONSORING MILITARY ACTIVITY	
		Naval Postgraduate School Monterey, California 93940	
13. ABSTRACT			
<p>The application of digital computers for the simulation of physical systems has become widespread. This paper describes a program designed to measure the frequency response of the simulation model of a linear system.</p>			

14

KEY WORDS

LINK A

LINK B

LINK C

ROLE

WT

ROLE

WT

ROLE

WT

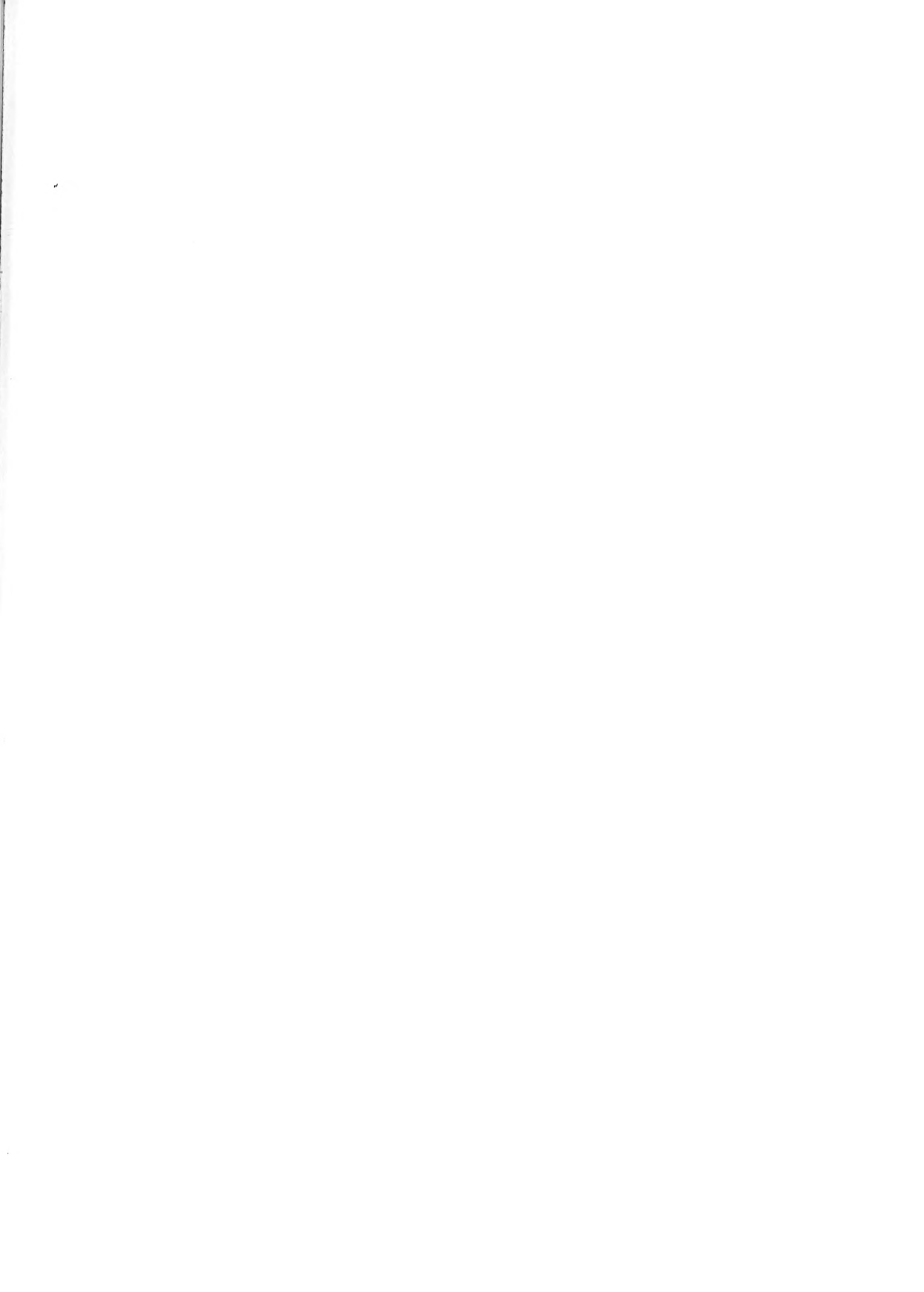
Frequency response

Computer simulated system

Digital simulation

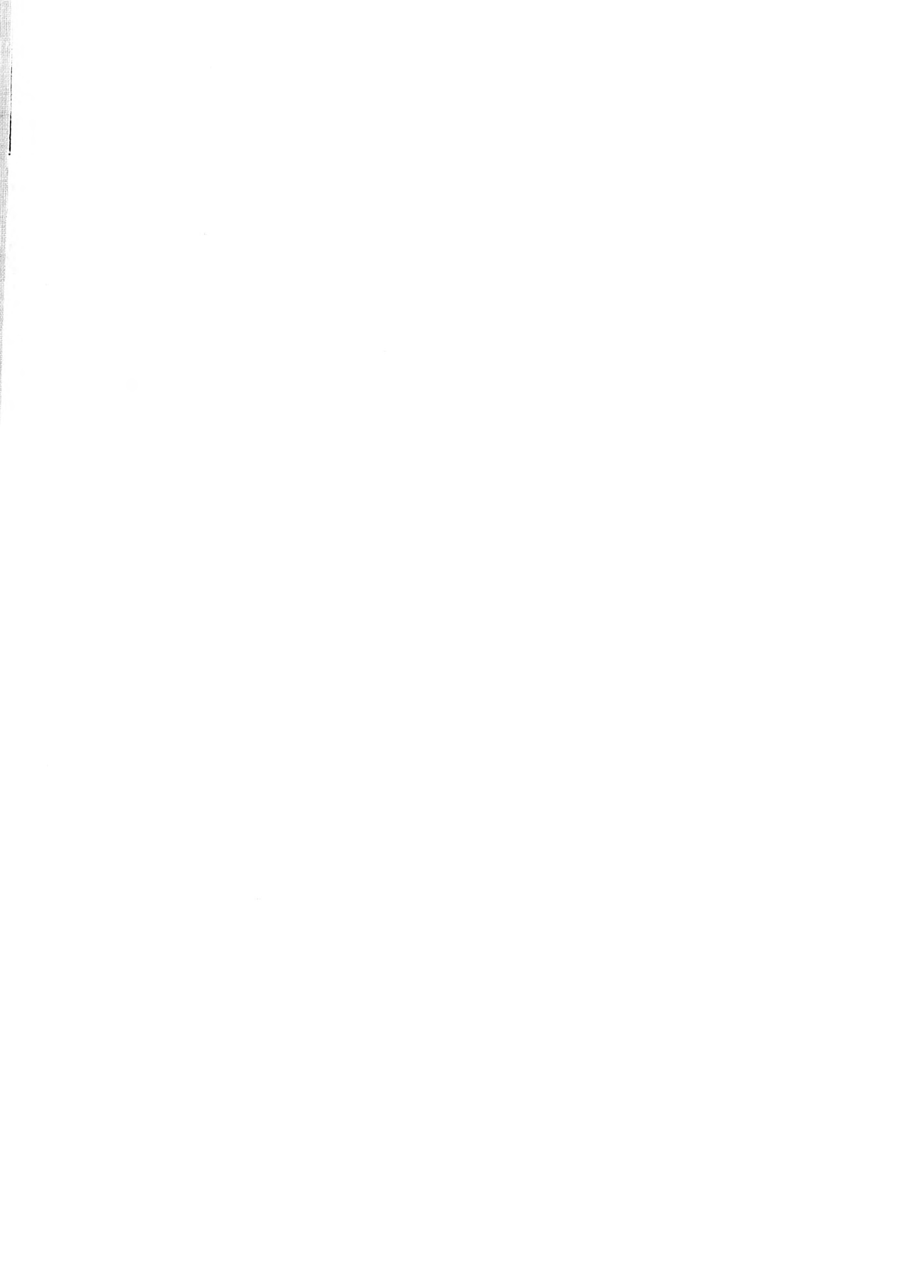












thesA3793

A digital computer program for determini



3 2768 000 98967 7

DUDLEY KNOX LIBRARY